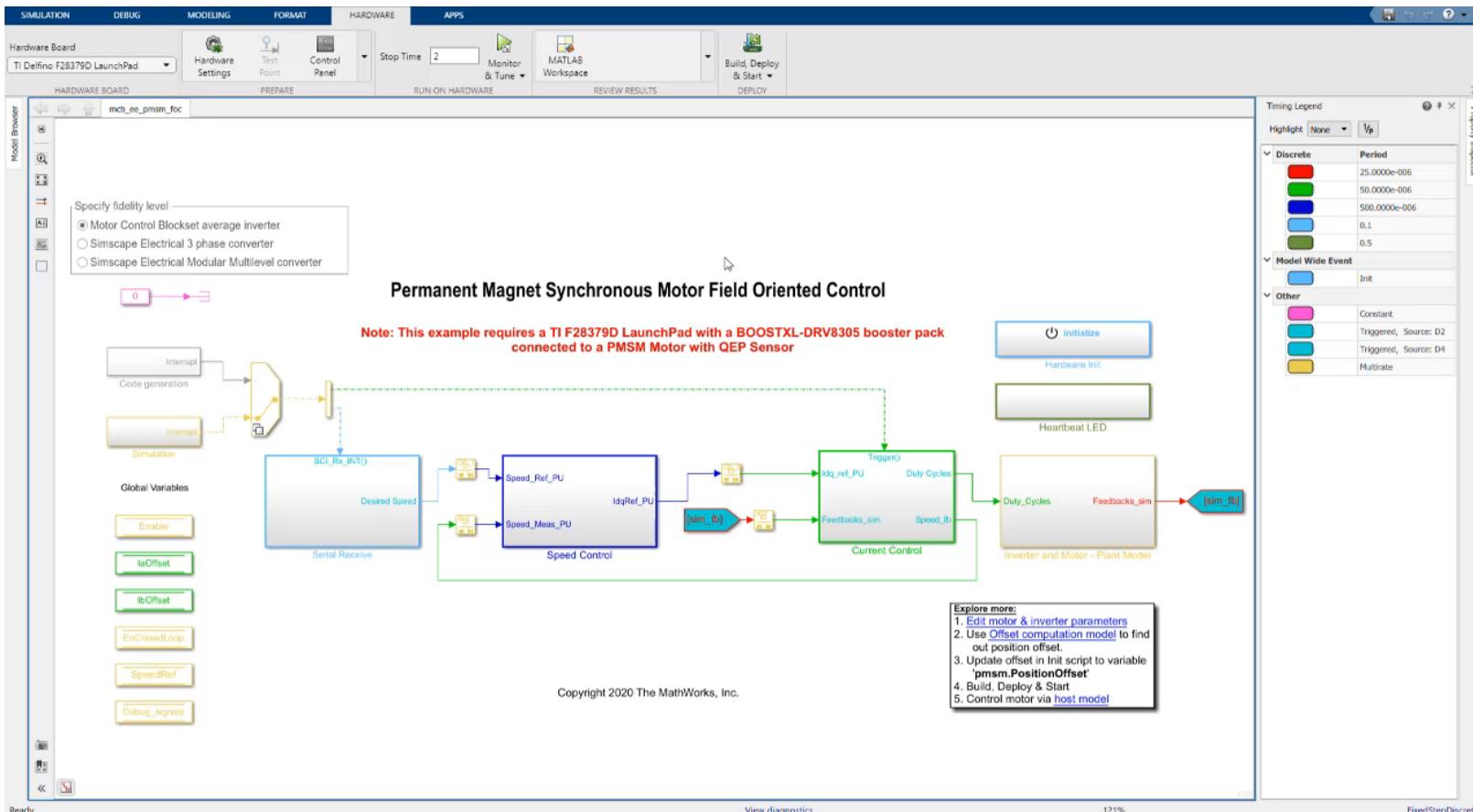


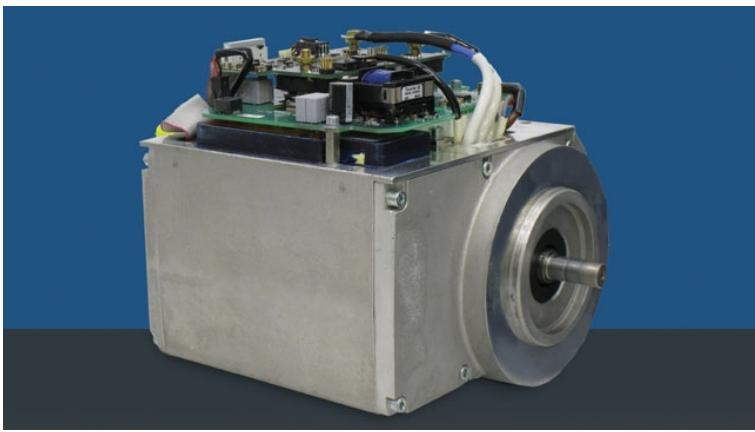
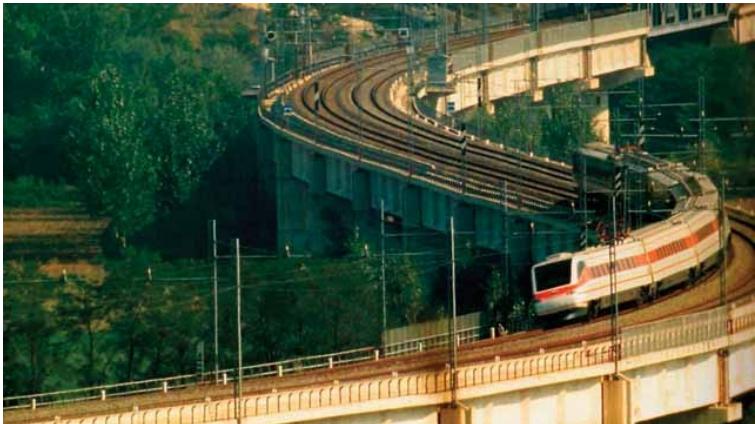
Deploying Motor Control Algorithms on NXP S32K from Simulink®

Brian McKay
Technical Marketing

We will spin a brushless motor using Simulink and Model-Based Design



Brushless motors are everywhere



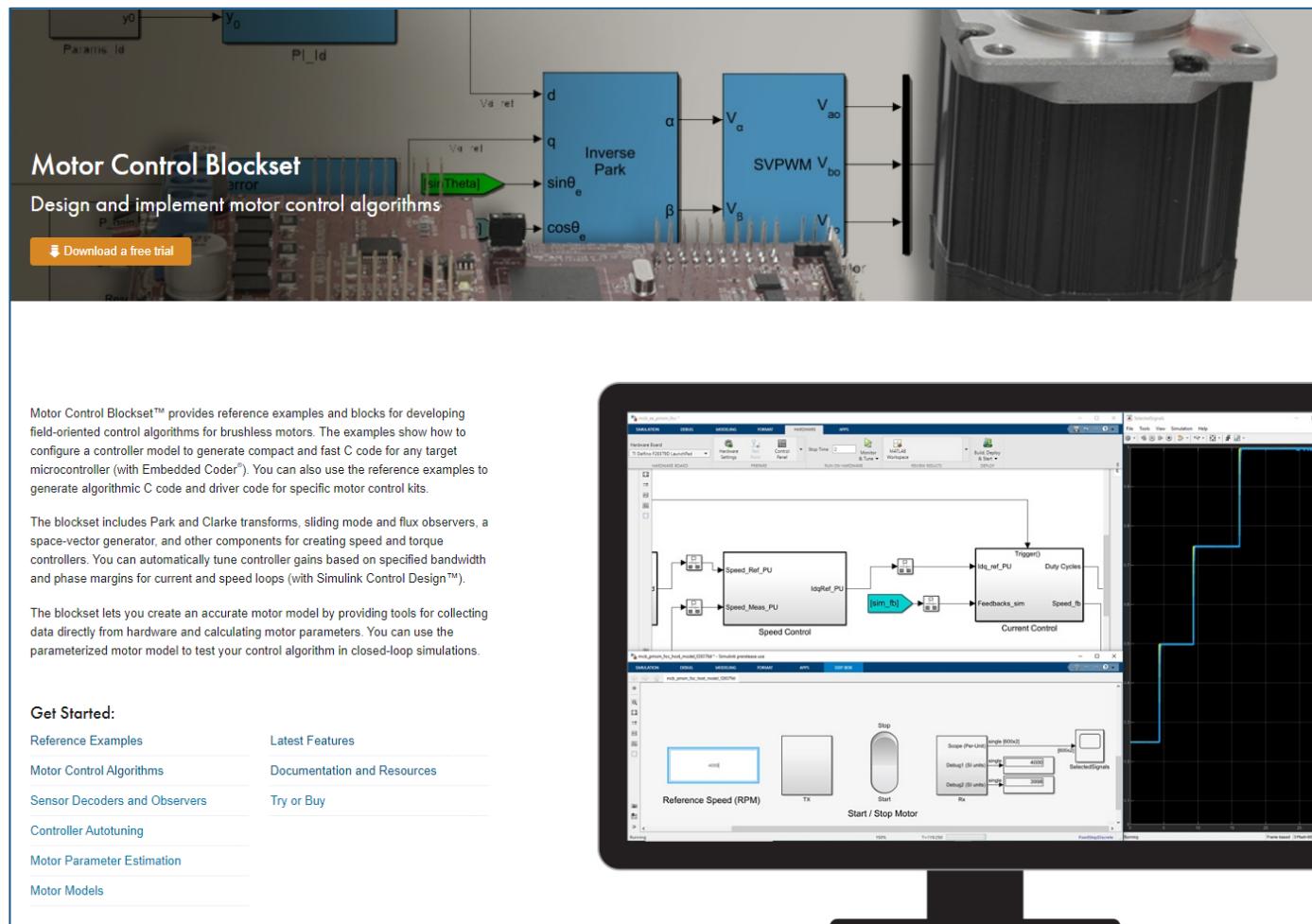
Why Simulink for motor control?

- Verify control algorithm with desktop simulation
- Generate compact and fast code from models
- Minimize development time using reference examples

Customers routinely report 50% faster time to market

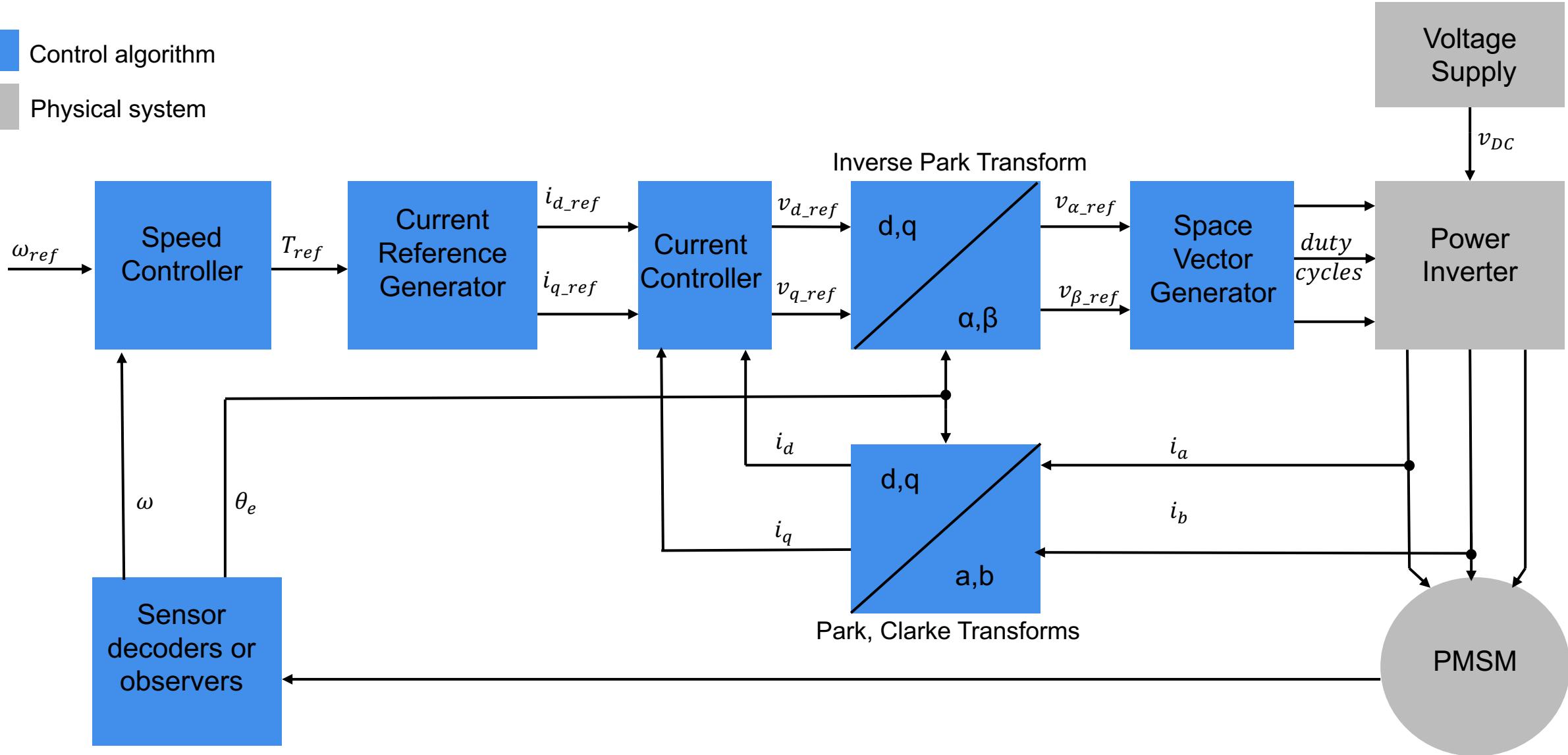
Motor Control Blockset simplifies the workflow

- Control blocks optimized for code generation
- Sensor decoders and observers
- Motor parameter estimation
- Controller autotuning
- Reference examples

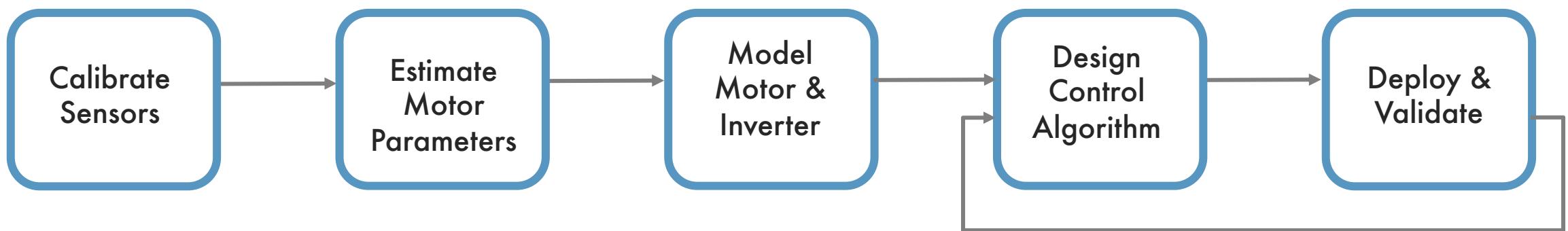


Brushless motors require complex algorithms – field-oriented control

 Control algorithm
 Physical system

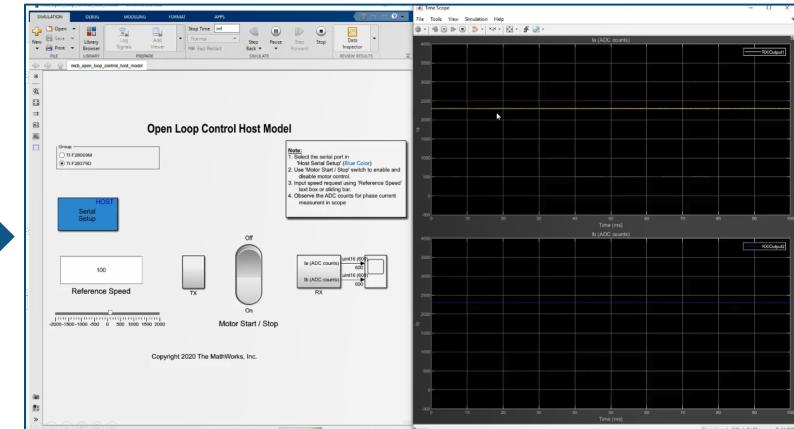
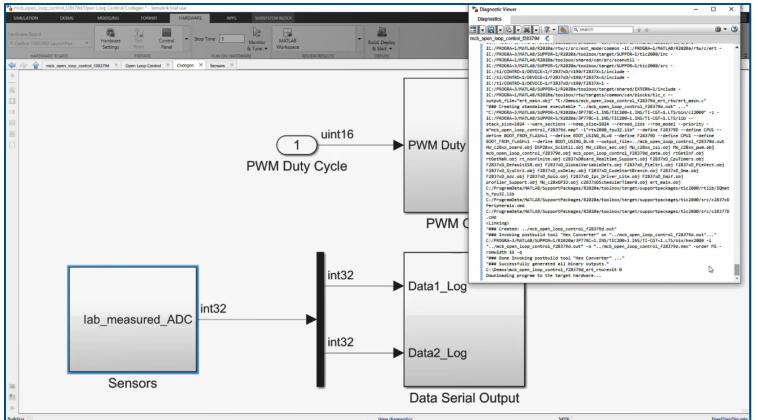
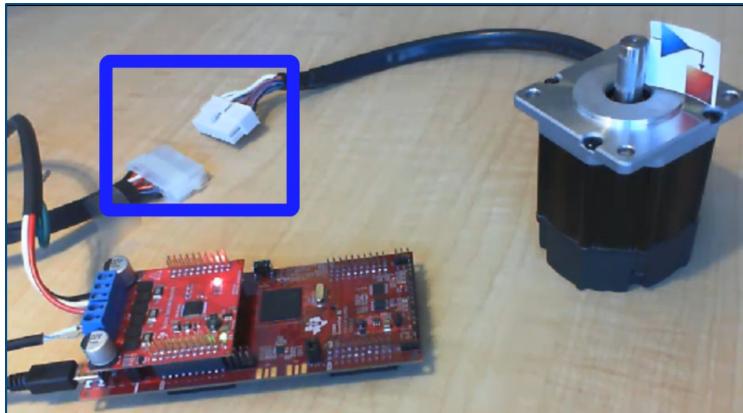
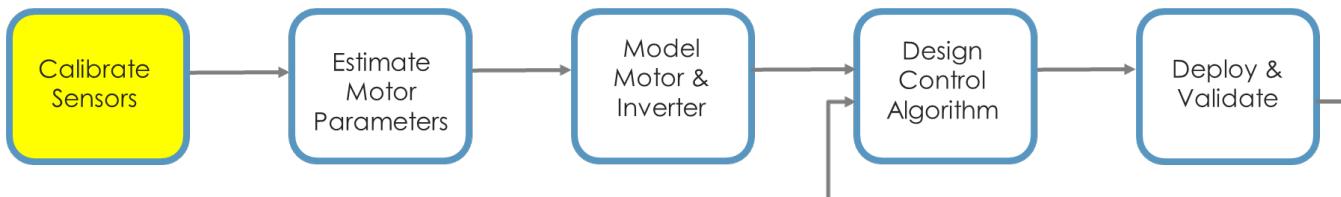


Workflow for implementing field-oriented control



Sensor calibration

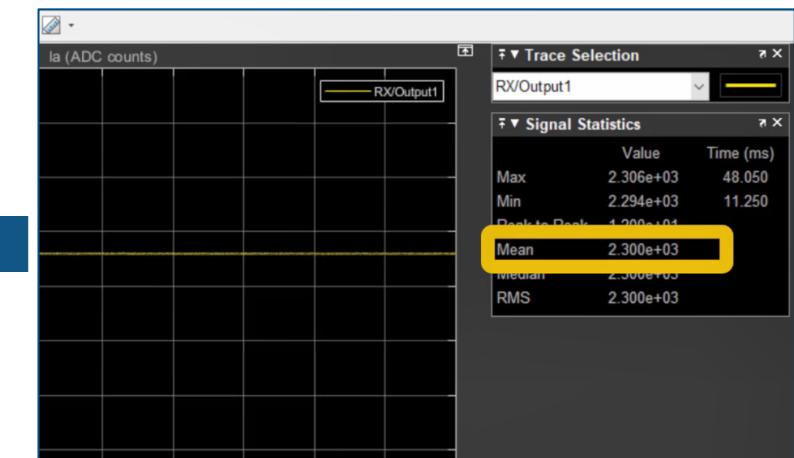
- Calibrate ADC offsets



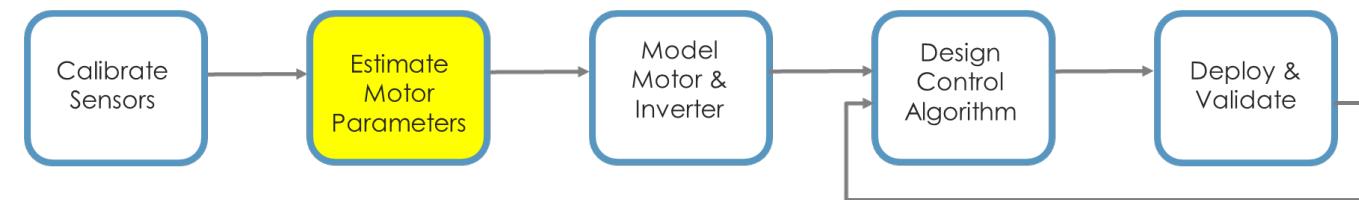
```

case 'BoostXL-DRV8305'
    inverter.model = 'BoostXL-DRV8305'; % Model
    inverter.sn = 'INV_XXXX';
    inverter.V_dc = 24; %V
    inverter.I_max = 19.3; %Amps
    inverter.I_trip = 10; %Amps
    inverter.Rds_on = 2e-3; %Ohms
    inverter.Rshunt = 0.007; %Ohms
    inverter.MaxADCCnt = 4095; %Counts
    inverter.CtSensAOffset = 2300; %Counts
    inverter.CtSensBOffset = 2303; %Counts
    inverter.ADCGain = 1; % Gain
  
```

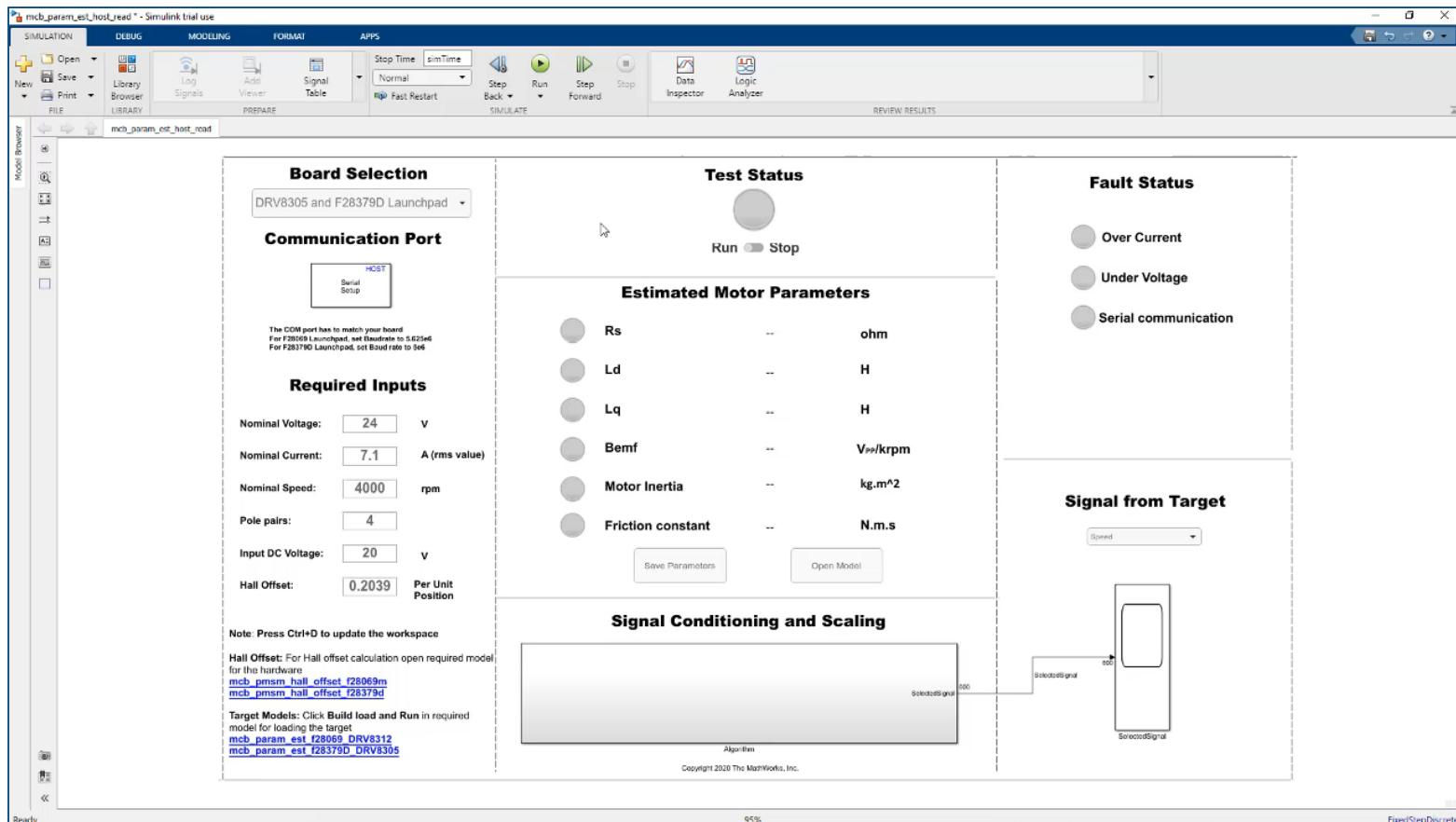
A code snippet from a MATLAB script defining parameters for a 'BoostXL-DRV8305' inverter model. It includes variables for model name, serial number, supply voltage, maximum current, trip current, on-resistance, shunt resistance, maximum ADC count, and sensor offsets. The 'inverter.CtSensAOffset' value is highlighted with a yellow box.



Parameter estimation

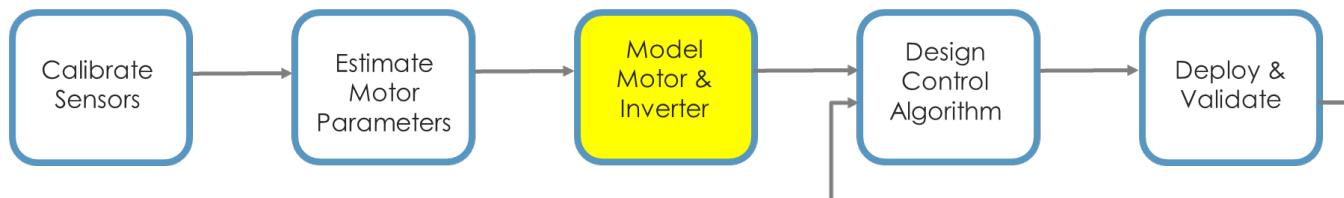


- Instrumented tests running on the target
- Host model to start and control parameter estimation

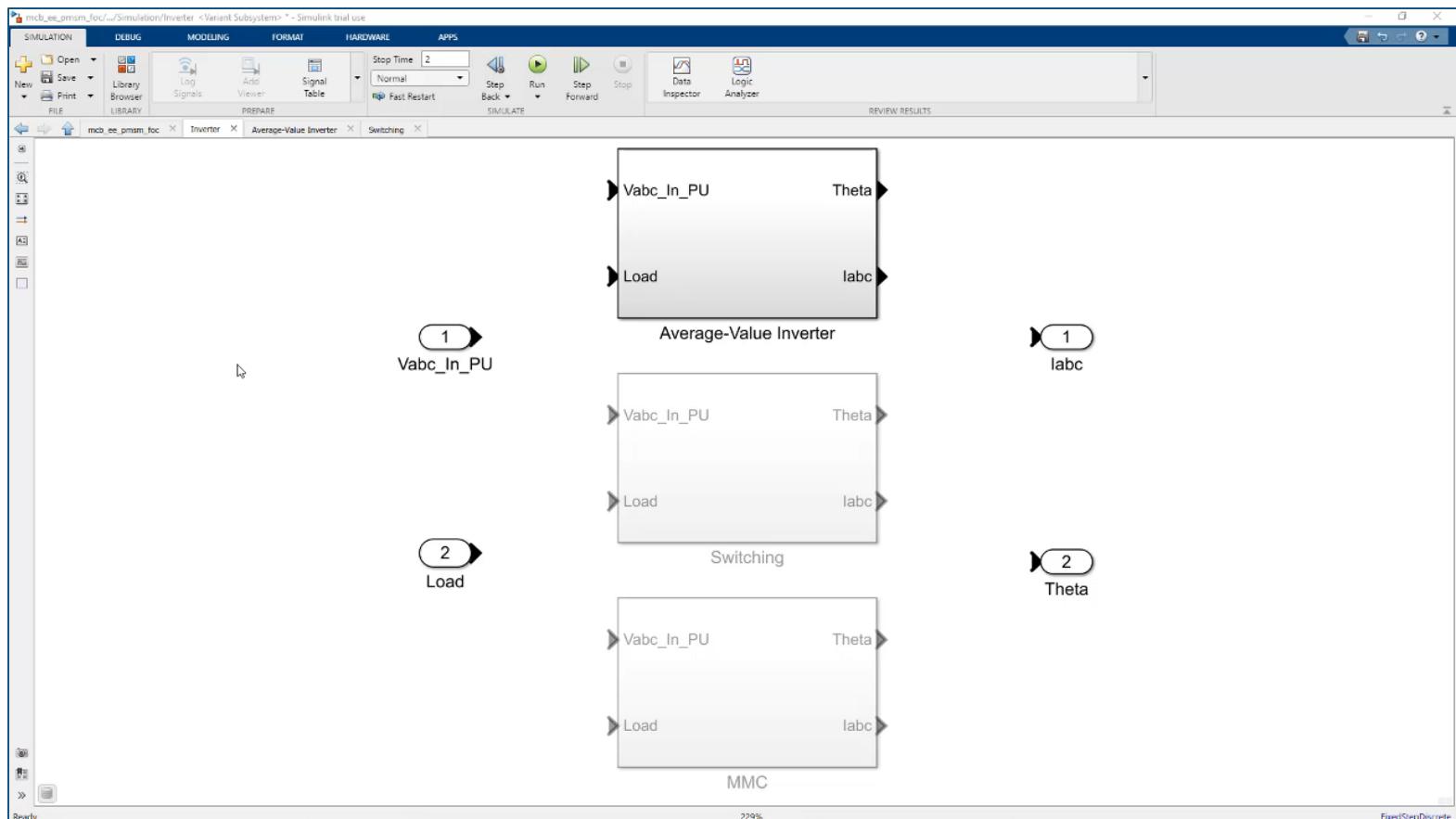


* Parameter estimation is supported by quadrature encoder, Hall sensors, or sensorless flux observer

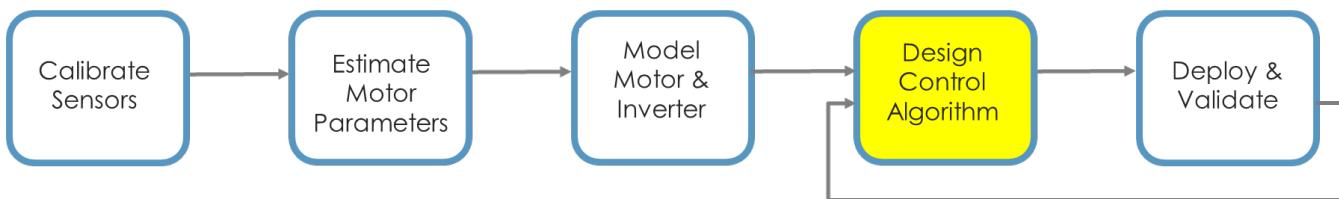
Modeling motor and inverter



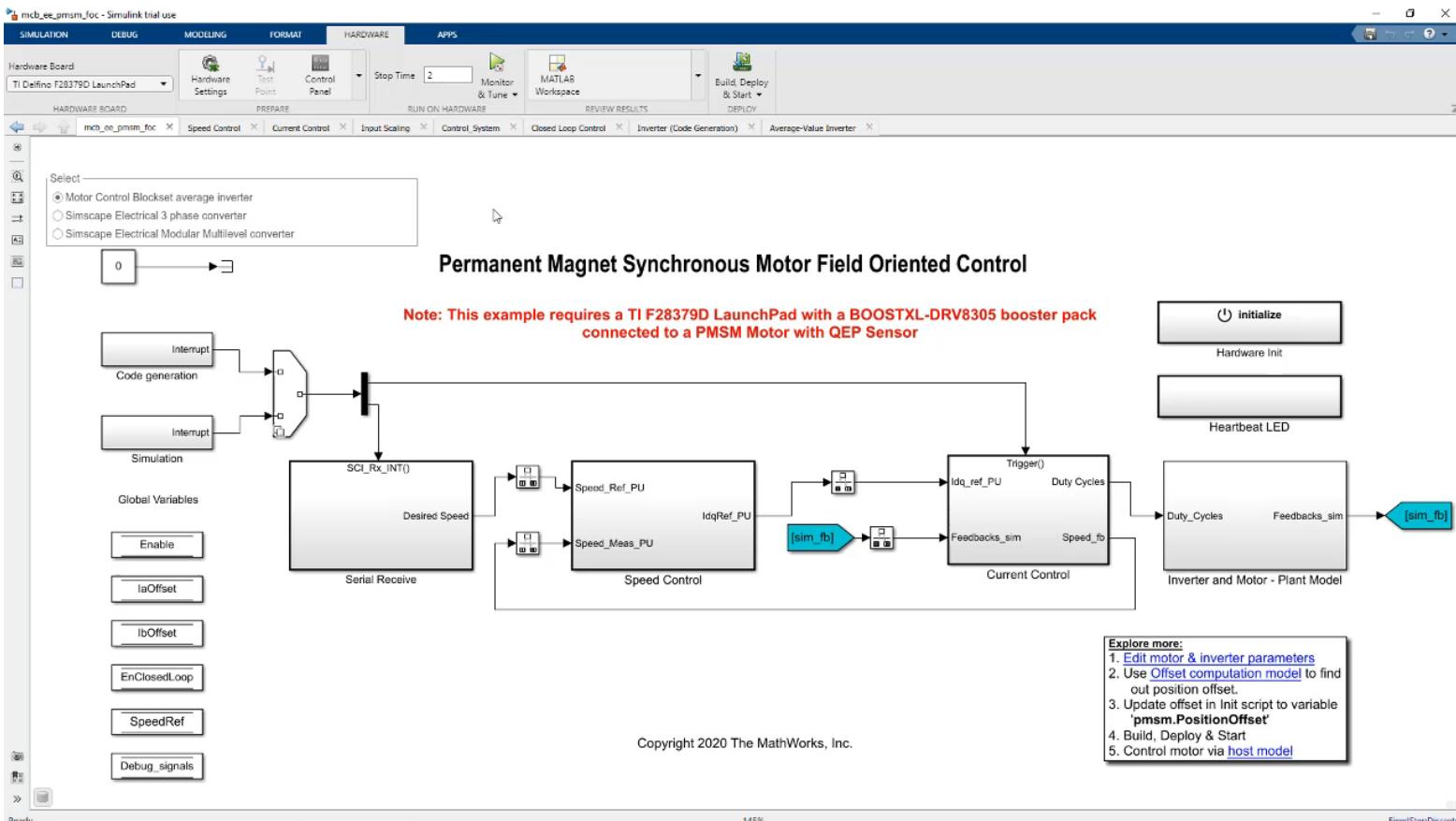
- Use linear lumped-parameter motor model
- Model inverter as an average-value inverter or model switching with Simscape Electrical



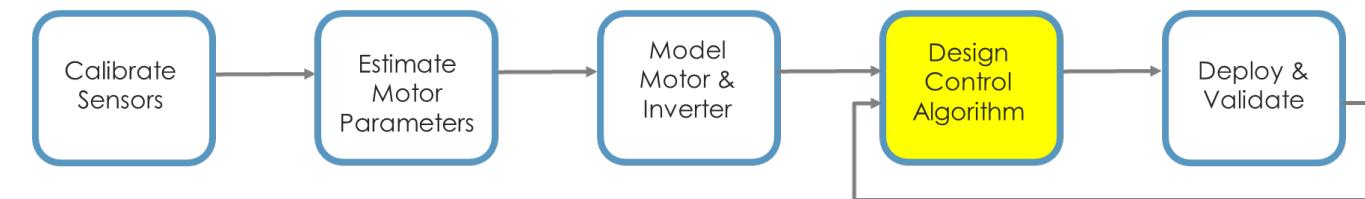
Control algorithm design



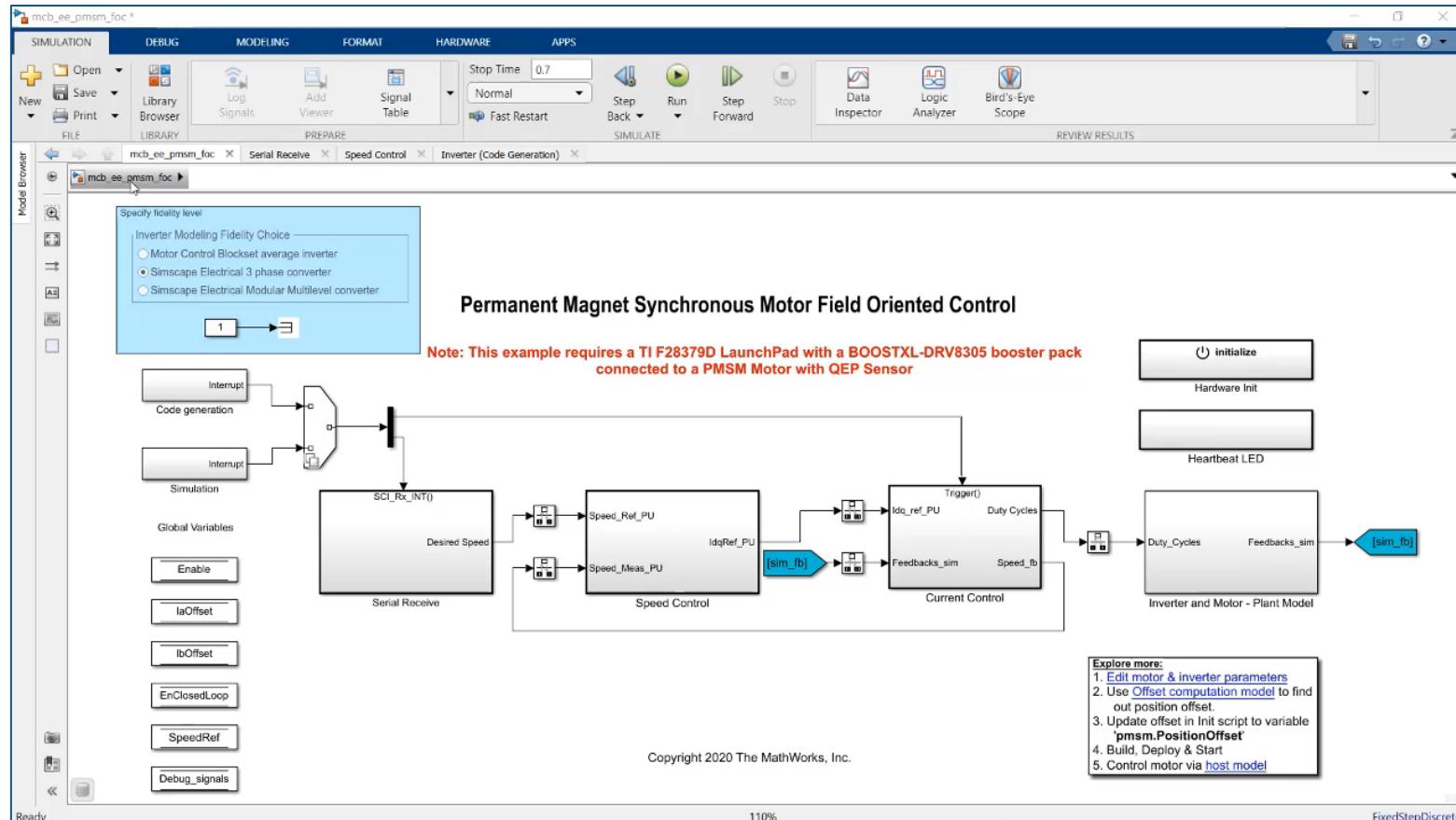
- Model field-oriented control algorithm
- Model sensor decoders or sensorless observers
- Tune loop gains
- Verify in closed-loop simulation



Control algorithm design

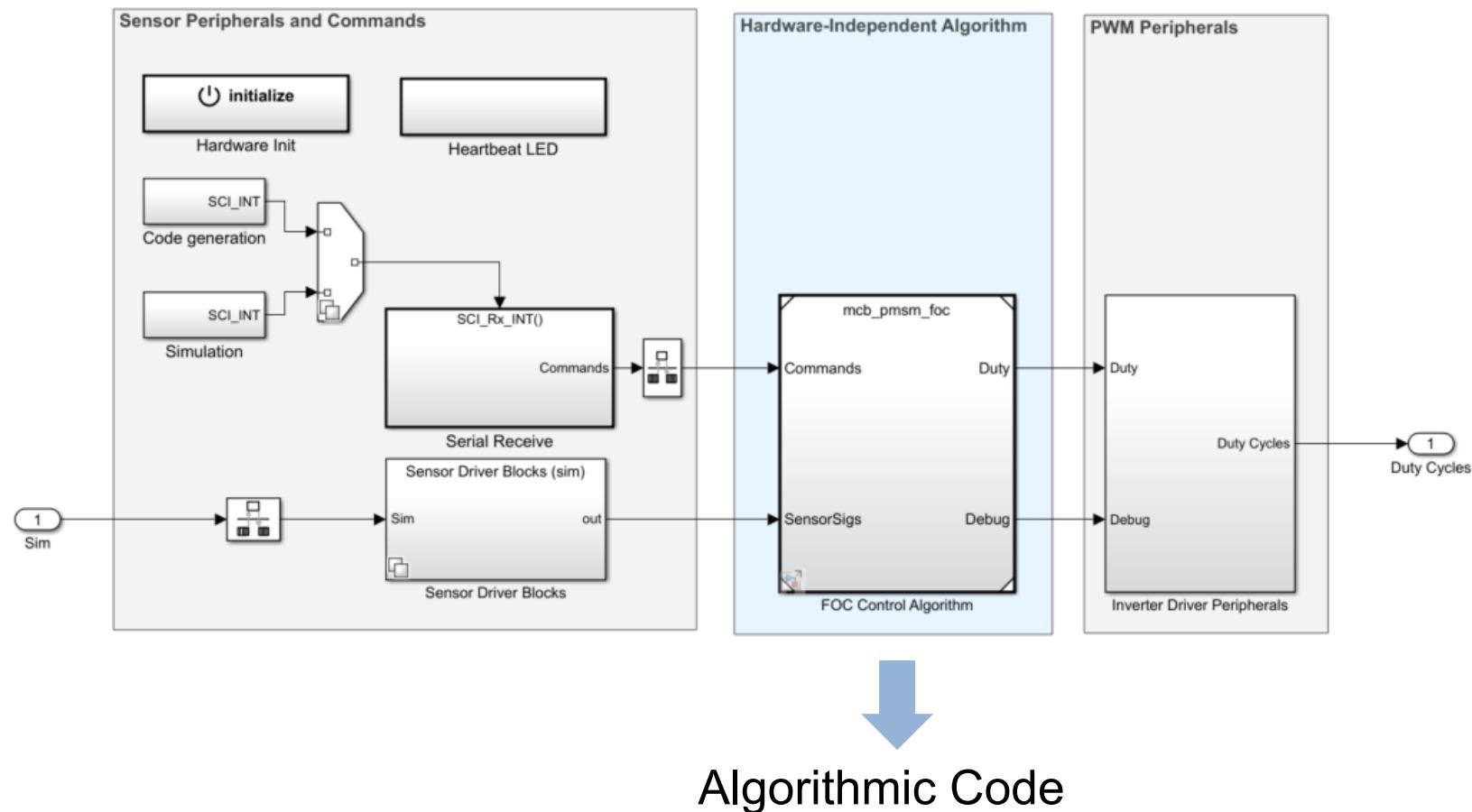
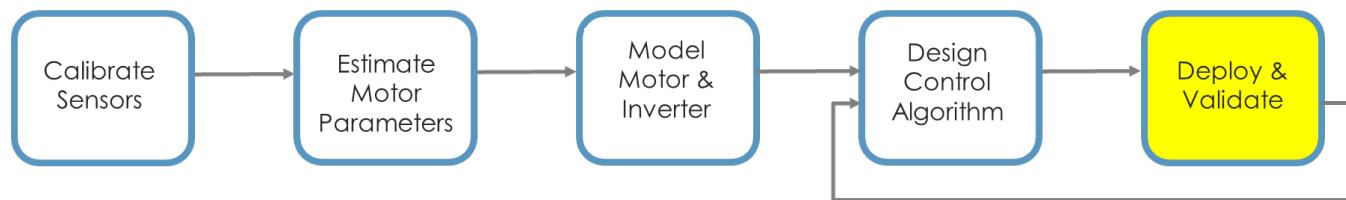


- Model field-oriented control algorithm
- Model sensor decoders or sensorless observers
- Tune loop gains
- Verify in closed-loop simulation

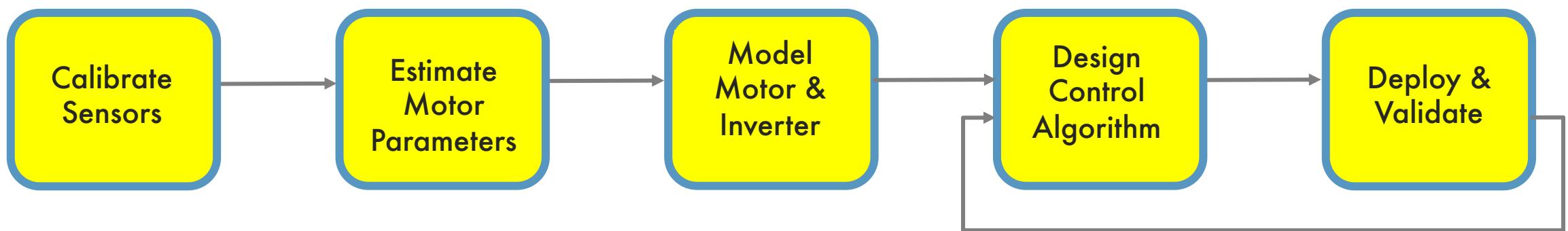


Deployment

- Target NXP S32K MCU with automatically generated C code
- Use provided example to partition the model into algorithmic and hardware-specific parts
- Validate on hardware



Workflow for implementing field-oriented control



DEPLOYING MOTOR CONTROL ALGORITHMS ON NXP S32K FROM SIMULINK

Daniel Popa
MBDT Product Manager & Architect
DECEMBER 2020



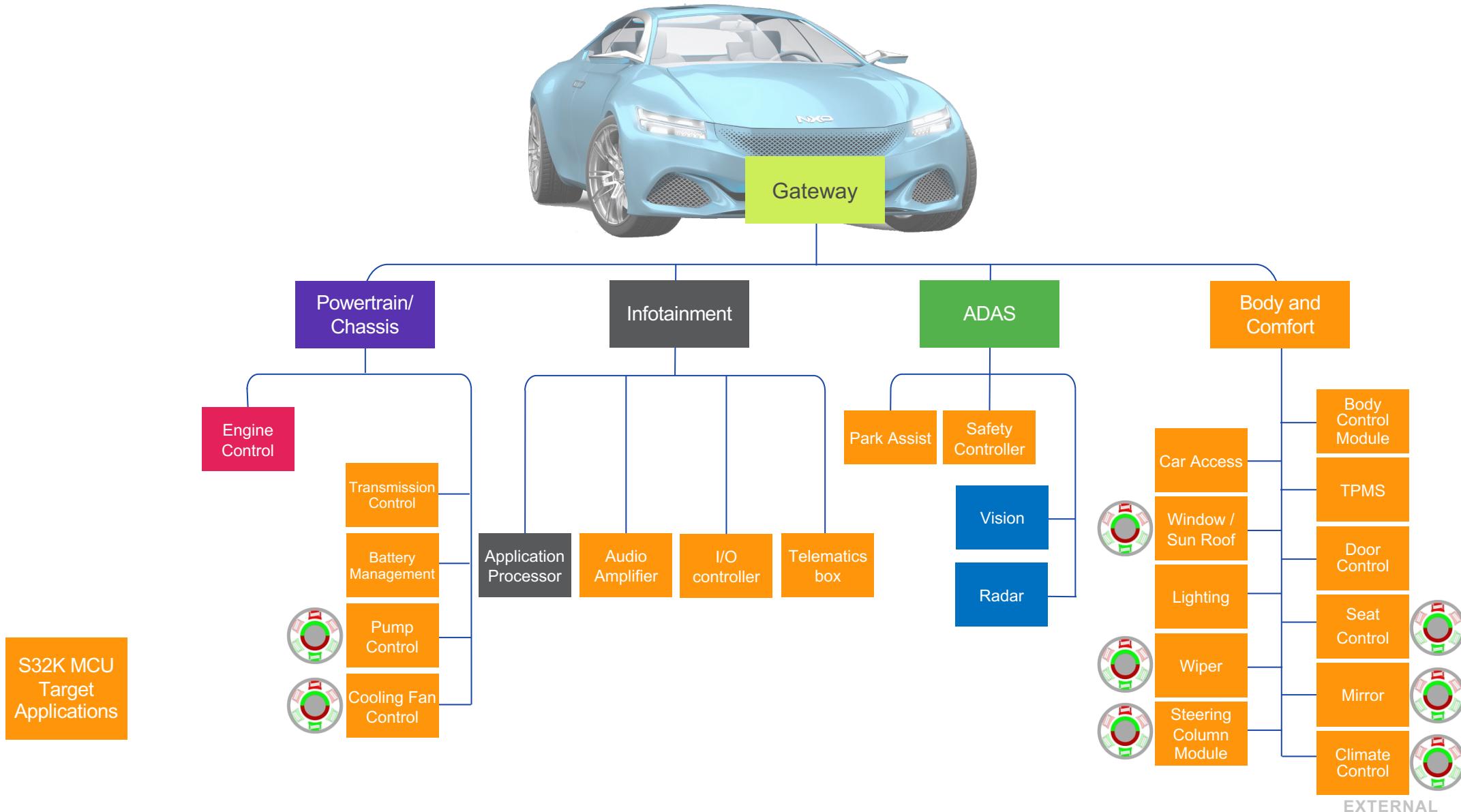
SECURE CONNECTIONS
FOR A SMARTER WORLD

EXTERNAL

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.



S32K – SOLUTIONS FOR EDGE NODES



NXP MOTOR CONTROL SOLUTIONS



[MCSpte1ak144](#)



[Mtrcktsp5744p](#)



[Mtrcktdbn5643l](#)



[Mcsxte2bk142](#)



[Mtrcktsp5643l](#)



[Mtrcktdps5643l](#)

NXP MOTOR CONTROL SOLUTIONS



[MCSpte1ak144](#)



[Mtrcktsp5744p](#)



[Mtrクトdbn5643l](#)



[Mcsxte2bk142](#)

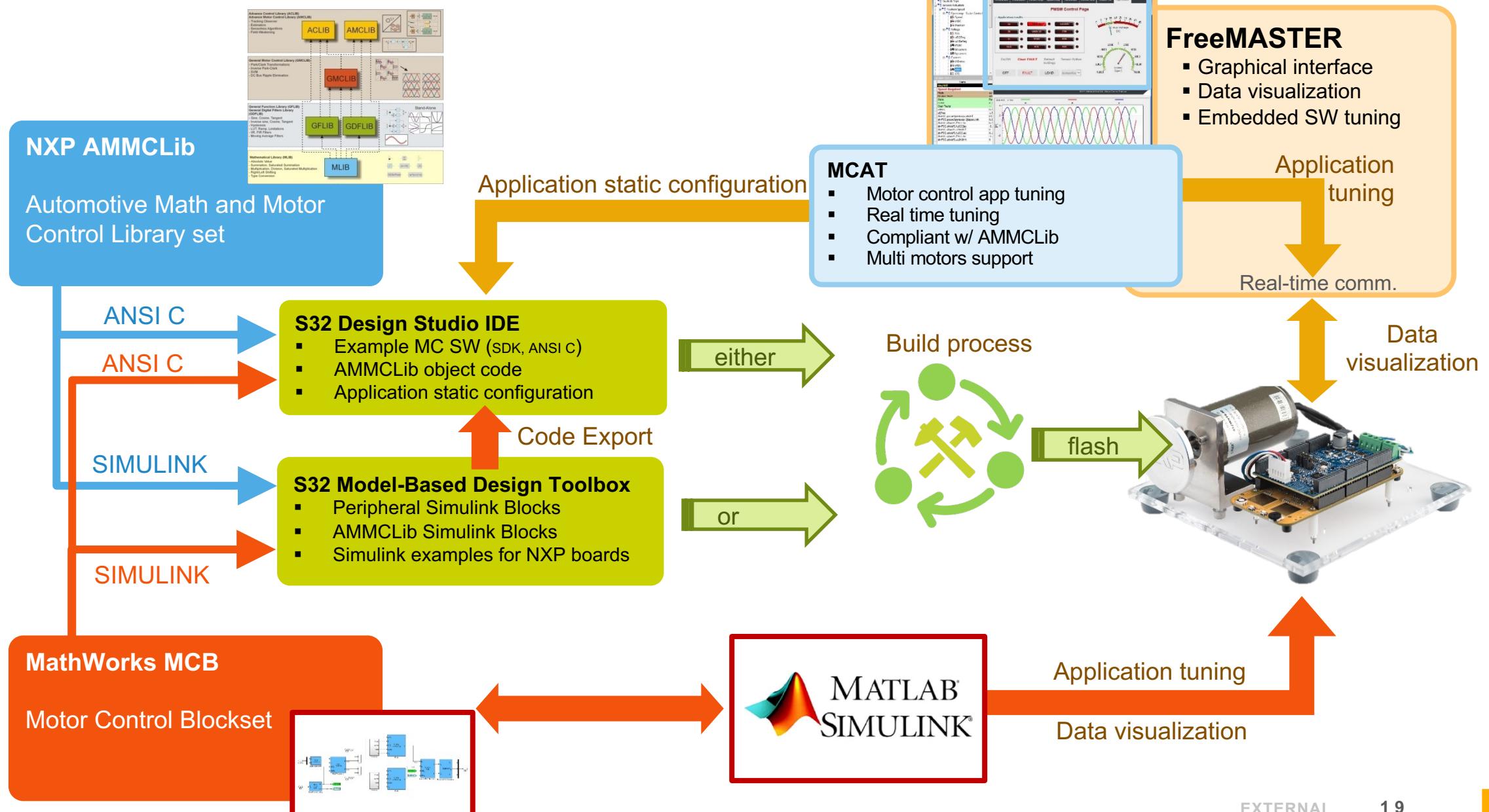


[Mtrクトsp5643l](#)



[Mtrクトdps5643l](#)

NXP TOOLCHAIN SW DEVELOPMENT FLOW



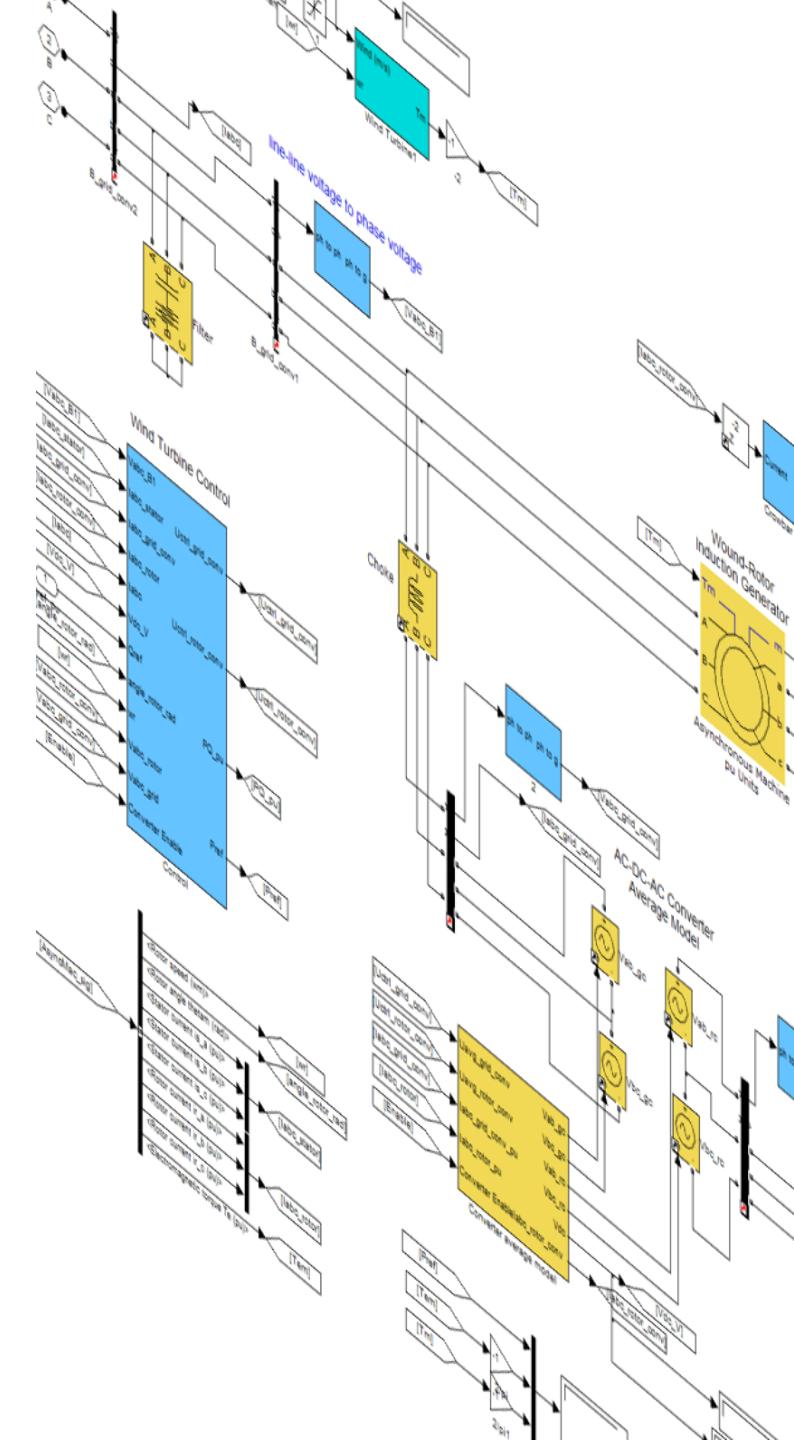
PMSM FOC with MCB on MCSPT1AK144



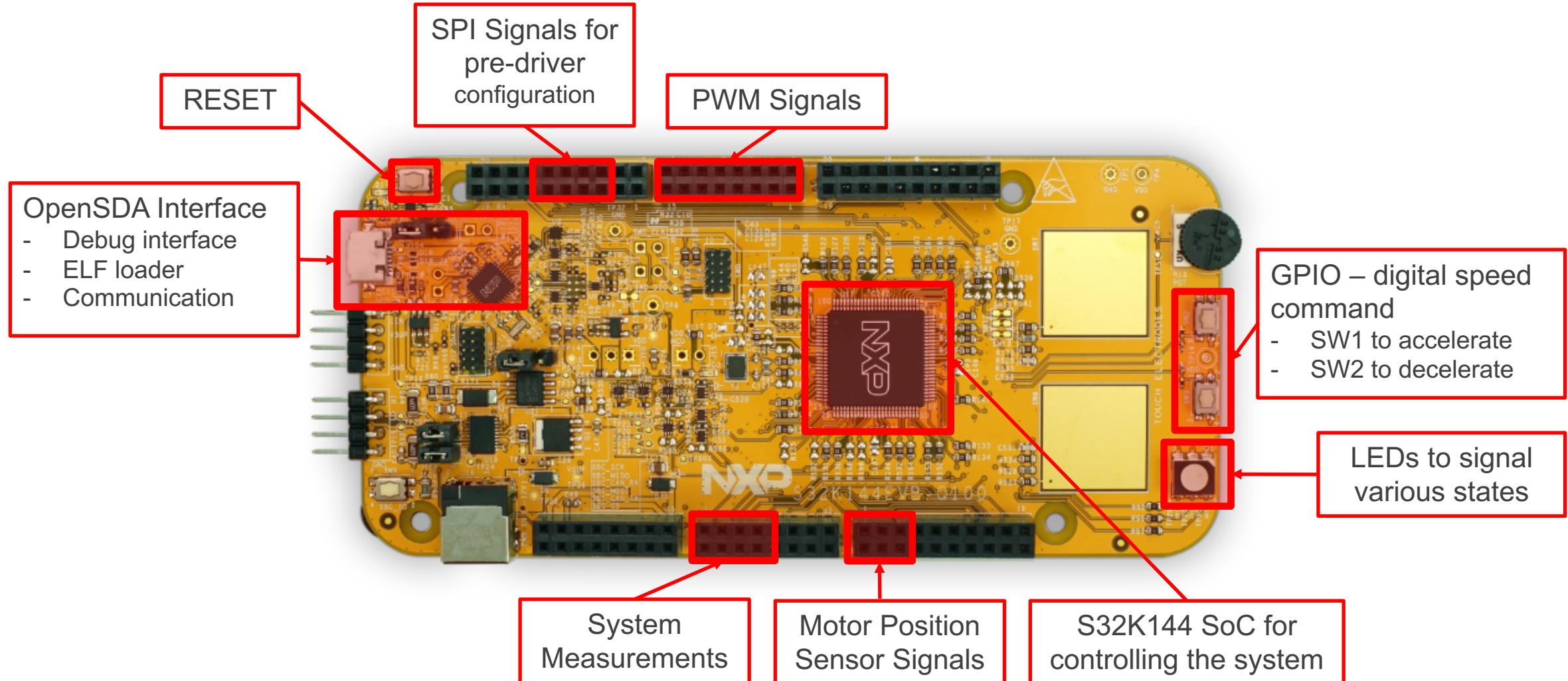
SECURE CONNECTIONS
FOR A SMARTER WORLD

EXTERNAL

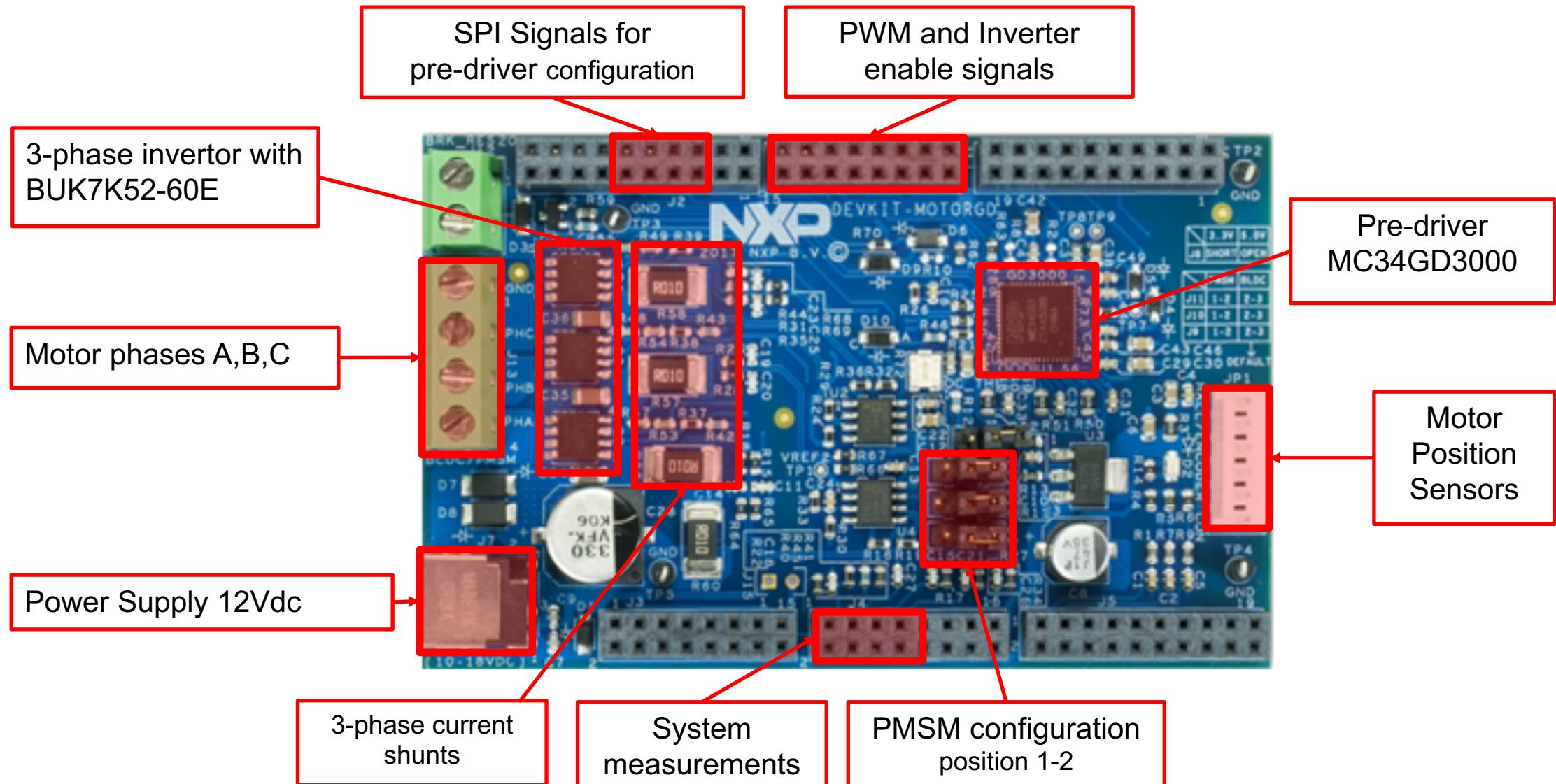
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.



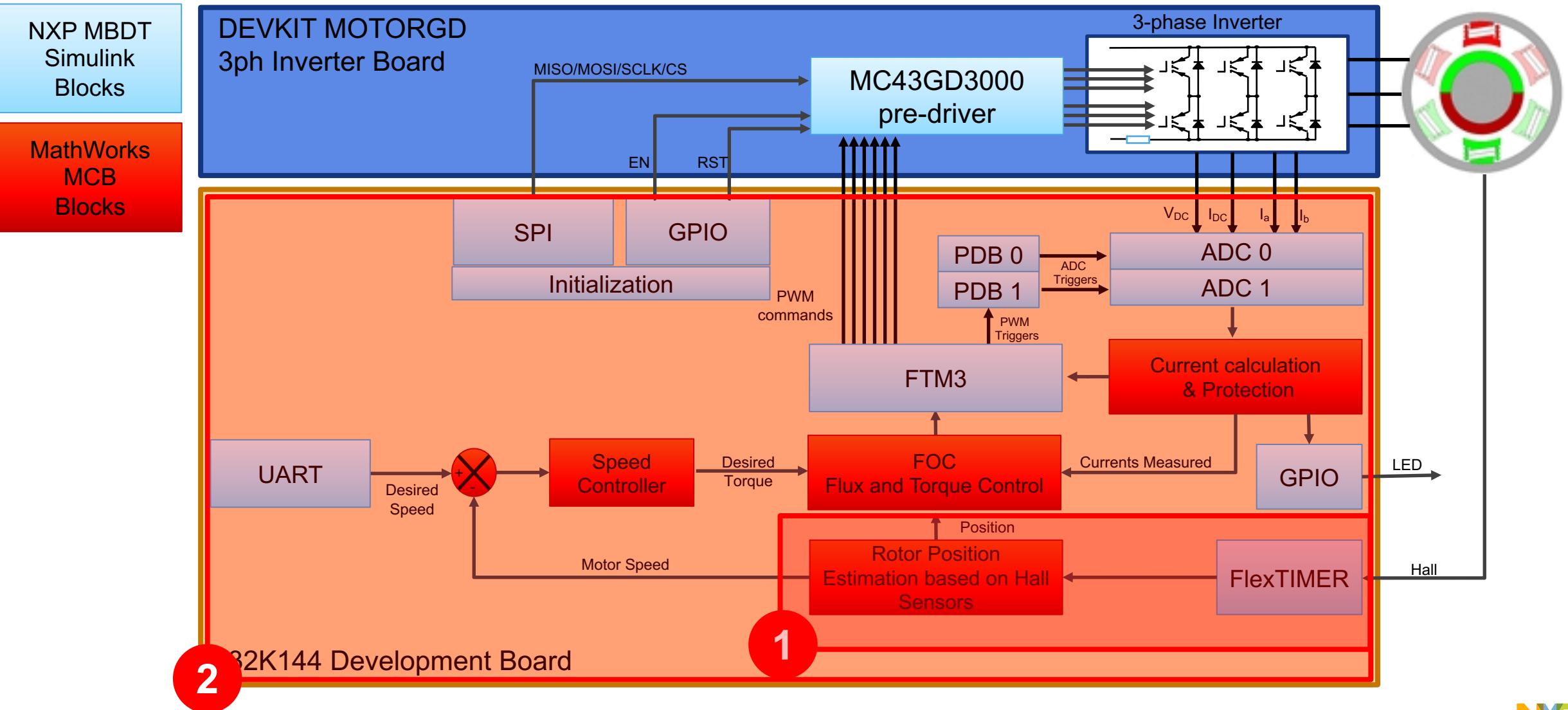
S32K144EVB – PERIPHERALS USED IN THE DEMO



DEVKIT MOTORGD – PERIPHERALS USED IN THE DEMO



PMSM FOC BLOCK DIAGRAM



mcb_pmsm_hall_offset_S32K144EVB_NXP - Simulink

SIMULATION DEBUG MODELING FORMAT APPS C CODE X

Custom Target ▾ Quick Start C/C++ Code Advisor ▾ Settings ▾ Code Interface ▾ PREPARE Code for mcb_pmsm_hall_offset_S32K144EVB_NXP Build ▾ View Code Open Report ▾ Remove Highlighting Verify Code ▾ Share ▾ RESULTS VERIFY SHARE ▾ SHARE ▾

Tools

mcb_pmsm_hall_offset_S32K144EVB_NXP

mcb_pmsm_hall_offset_S32K144EVB_NXP ▾

Offset Computation with Hall sensor

Explore more:

1. [Edit motor & inverter parameters](#)
2. Build, Deploy & Start
3. Control motor via [FreeMaster project](#)
4. Connect using 57600 bauds
5. Set "Enable" variable to 1 in FreeMaster to start the test

Target : S32K144 64KBSRAM NXP
 Package : 100-LQFP
 System clock : 80 MHz
 XTAL clock : External 8 MHz
 Compiler : GCC
 Target Type : FLASH
 Download Code after build : (OpenSDA D)
 Step Tick Interrupt Priority : 10

MBD_S32K1xx_Config_Information

Hardware Init

GlobalHallState
 HallStateChangeFlag
 GlobalSpeedCount
 GlobalSpeedValidity
 GlobalDirection
 Enable
 FAULT

FTM Hall Sensor NXP Trigger
 Module : 2
 Channel : 1
 Edge-sensitive mode : Rising or Falling
 Prescale factor : 128
 FTM Counter

function()
 1

Hall Sensors

HW_Events

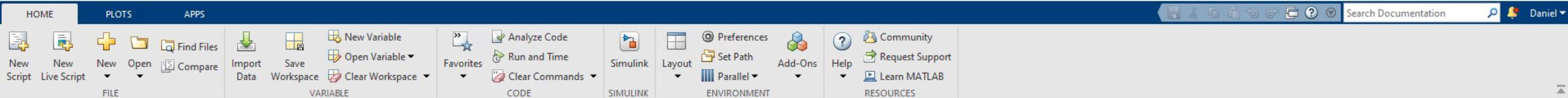
FcnCall
 ADC_res → lab_meas_ADC

Trigger()
 Offset Calculation

Note: This example requires a NXP MCSPT1AK144 Motor Control Kit

Model Data Editor Code Mappings - C

Ready 180% FixedStepDiscrete



Current Folder

Name
FreeMASTER_Control
mcb_pmsm_foc_hall_S32K144EVB.mdl
mcb_pmsm_foc_hall_S32K144EVB.pmpx
mcb_pmsm_foc_hall_S32K144EVB_data.m

Command Window

>>

Workspace

Name	Value
------	-------

mcb_pmsm_foc_hall_S32K144EVB_data.m (Script)

```
*****
% Parameters needed for Offset computation are
% Set PWM Switching frequency
% Set Sample Times
% Set data type for controller & code-gen
% System Parameters // Hardware parameters
% Parameters below are not mandatory for offset computation
% Derive Characteristics
```

GETTING HELP

[MBDT Online Community Examples & Help](#)

[Home](#) / [NXP Model-Based Design Tools](#) / [NXP Model-Based Design Tools](#)

NXP Model-Based Design Tools

Model-Based Design Tools for Matlab and Simulink Support

S32K
• Tutorials
• Videos

MPC57xx
• Tutorials
• Videos

Other Solutions
• Tips and Tricks

The Model-Based Design Toolbox provides an integrated development environment and toolchains for configuring and generating all of the necessary software automatically. [Learn more](#).

Discussions

Post Questions

Top Kudoed Authors

S32K1 MCUs

MC33771 & MC33772 EVB Data reading issue

MC33771 Balancing Logic

MC33771 Status in MBDT

FDI Rx Data Size and Type mismatch - What does it mean?

CAN FD in S32K1 MCUs

S32K14x LPSPI setting module "Device Amt" parameter

Latest articles

Model-Based Design Toolboxes for S12X/Mx Automotive ...

BMS & MBDT - MC33771&MC33772 SPI communication via ...

BMS & MBDT - MC33771&MC33772 SPI communication via ...

BMS & MBDT - Designing Battery Management System w...

[MBDT home page](#) www.nxp.com/mbdt

[OVERVIEW](#) [DOCUMENTATION](#) [DOWNLOADS](#) [DEVELOPMENT TOOLS](#) [TRAINING & SUPPORT](#)

Overview

Jump to Overview & Features Supported Devices Target Applications System Requirements

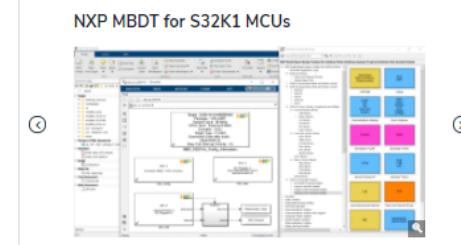
The NXP Model-Based Design Toolbox (MBDT) is a comprehensive collection of tools that plug into the MATLAB® and Simulink® model-based design environment to support fast prototyping, verification, and validation on NXP microcontroller-based real targets. The free-of-charge NXP MBDT includes an integrated Simulink-embedded target supporting NXP MCUs for direct rapid prototyping and built-in support for software development, code generation, and simulation.

Features

Comprehensive free-of-charge build toolchain for embedded applications, targeting NXP microcontrollers. Generate code capabilities for standalone application with direct download to target support. Out-of-the-box applications for a wide set of MCU peripherals. On-demand reporting of functions and tests.

DOWNLOAD

NXP MBDT for S32K1 MCUs



S32K1: S32K1 General Purpose MCUs

MPC574xP: Ultra-Reliable MPC574xP MCU for Automotive & Industrial Safety Applications

MPC574xL: Ultra-Reliable MPC574xL MCU for Automotive & Industrial Radar Applications

MPC574xK: Ultra-Reliable MPC574xK MCU for Automotive & Industrial Motor Control Applications

S12ZVM: S12ZVM Mixed-Signal MCU for Automotive & Industrial Motor Control Applications

MC56F8270x: MC56F8270xx; MC56F8280xx; MC56F8270x Digital Signal Controllers

IMX-RT1010: IMX RT1010 Crossover MCU with Arm® Cortex®-M7 core

IMX-RT1060: IMX RT1060 Crossover MCU with Arm® Cortex®-M7 core

IMX-RT1064: IMX RT1064 Crossover MCU with Arm® Cortex®-M7 core

Supported Devices

EXTERNAL

26





SECURE CONNECTIONS
FOR A SMARTER WORLD